РУСЕНСКИ УНИВЕРСИТЕТ КАТЕДРА

„АНГЕЛ КЪНЧЕВ“ „КСТ“

**КУРСОВА ЗАДАЧА**

по „Програмни езици“

на Диан Лъчезаров Лазаров

студент II курс, специалност КСТ, група 25Б, фак. № 233078

**Русе** Разработил: ...................................

**13.12.2024 г.** **/**Д. Лазаров**/**

Приел: ...................................

**/**доц. д-р Г. Георгиев**/**

**ЗАДАНИЕ**

Да се разработи елементарен векторен графичен редактор на програмния език Java.

* Графичният редактор да позволява избор на цвят за изчертаване и евентуално на цвят за запълване и да поддържа следните графични примитиви: **незапълнен четириъгълник, незапълнен овал, линия.**
* Изборът на примитив, който ще се въвежда, да става чрез **списък (java.awt.List / javax.swing.JList)**.
* Потребителският интерфейс и самите графични примитиви да са разположени в **един и същ прозорец.**
* По време на работа с редактора обектите от графичните примитиви да се съхраняват в **ArrayList** (**java.util.ArrayList**).В потребителския интерфейс да се предвиди команда за извеждане на съдържанието му в конзолата.
* В програмата може да се използват различни класове по усмотрение на студента, но задължително за всеки графичен примитив трябва да има отделен клас. Всички тези класове трябва да реализират следния интерфейс:

**public interface DrawableShape {**

**public abstract void drawShape(Graphics g);**

**// методът изчертава графичен примитив**

**}**

* За базов тип на **колекцията** трябва да се използва интерфейсът **DrawableShape**, а като елементи да се съхраняват обекти от класовете за отделните фигури.
* Как се задават фигурите (например четириъгълник може да бъде зададен по две точки или по точка, ширина и височина) и как се въвеждат стойности (например чрез работа с мишката, както в комерсиалните редактори, или чрез въвеждане на числа в някакъв потребителски интерфейс), е по избор на студента.
* Изборът на пакет за графика/GUI между **java.awt** и **javax.swing** е също според предпочитанията на студента.

**КЛАС-ДИАГРАМА**

+ drawShape(g: Graphics):: void

+ toString():: String

+ paint(g: Graphics):: void

+ FindMin(a: int, b: int):: int

+ FindMod(a: int):: int

Rectangle

- x: int

- y: int

- width: int

- height: int

- color: Color

+ drawShape(g: Graphics):: void

+ toString():: String

Oval

- x: int

- y: int

- width: int

- height: int

- color: Color

Line

- startX: int

- startY: int

- endX: int

- endY: int

- color: Color

+ drawShape(g: Graphics):: void

+ toString():: String

- shapeList: List

- shapes: ArrayList<DrawableShape>

- startX: int

- startY: int

- endX: int

- endY: int

- x: int

- y: int

- width: int

- height: int

- currentColor: Color

- chosenColor: Color

- selectedShape: String

GraphicsEditor

<<interface>>

DrawableShape

+ drawShape(g: Graphics):: void

**ЛИСТИНГ НА SOURCE КОДА**

**===DrawableShape.java===**

**import java.awt.\*;**

**public interface DrawableShape {**

**public abstract void drawShape(Graphics g);**

**}**

**===Line.java===**

**import java.awt.\*;**

**class Line implements DrawableShape {**

**int startX, startY, endX, endY;**

**Color color;**

**Line(int startX, int startY, int endX, int endY, Color color) {**

**this.startX = startX;**

**this.startY = startY;**

**this.endX = endX;**

**this.endY = endY;**

**this.color = color;**

**}**

**public void drawShape(Graphics g) {**

**g.setColor(color);**

**g.drawLine(startX, startY, endX, endY);**

**}**

**public String toString() {**

**return "Линия => startX: " + startX + ", startY: " + startY + ", endX: " + endX + ", endY: " + endY;**

**}**

**}**

**===Oval.java===**

**import java.awt.\*;**

**class Oval implements DrawableShape {**

**int x, y, width, height;**

**Color color;**

**Oval(int x, int y, int width, int height, Color color) {**

**this.x = x;**

**this.y = y;**

**this.width = width;**

**this.height = height;**

**this.color = color;**

**}**

**public void drawShape(Graphics g) {**

**g.setColor(color);**

**g.drawOval(x, y, width, height);**

**}**

**public String toString() {**

**return "Овал => x: " + x + ", y: " + y + ", width: " + width + ", height: " + height;**

**}**

**}**

**===Rectangle.java===**

**import java.awt.\*;**

**class Rectangle implements DrawableShape {**

**int x, y, width, height;**

**Color color;**

**Rectangle(int x, int y, int width, int height, Color color) {**

**this.x = x;**

**this.y = y;**

**this.width = width;**

**this.height = height;**

**this.color = color;**

**}**

**public void drawShape(Graphics g) {**

**g.setColor(color);**

**g.drawRect(x, y, width, height);**

**}**

**public String toString() {**

**return "Четириъгълник => x: " + x + ", y: " + y + ", width: " + width + ", height: " + height;**

**}**

**}**

**===GraphicsEditor.java===**

**import java.awt.\*;**

**import java.awt.event.\*;**

**import java.awt.List;**

**import java.util.ArrayList;**

**import javax.swing.JColorChooser;**

**class GraphicsEditor extends Frame {**

**List shapeList = new List();**

**ArrayList<DrawableShape> shapes = new ArrayList<DrawableShape>();**

**int startX, startY, endX, endY;**

**int x, y, width, height;**

**Color currentColor;**

**Color chosenColor;**

**String selectedShape;**

**GraphicsEditor() {**

**setTitle("Векторен графичен редактор");**

**setSize(900, 800);**

**setLocationRelativeTo(null);**

**// създаване на списък от примитиви**

**String Line = new String("Линия");**

**String Rectangle = new String("Четириъгълник");**

**String Oval = new String("Овал");**

**shapeList.add(Line);**

**shapeList.add(Rectangle);**

**shapeList.add(Oval);**

**shapeList.select(0);**

**// слушател за затваряне на прозореца**

**addWindowListener(new WindowAdapter() {**

**public void windowClosing(WindowEvent e) {**

**System.exit(0);**

**}**

**});**

**// бутон за извеждане на съдържанието на колекцията в конзолата**

**Button displayShapes = new Button("Отпечатване на съдържанието");**

**displayShapes.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**System.out.println("Изчертани примитиви:");**

**for (int i = 0; i < shapes.size(); i++) {**

**DrawableShape shape = shapes.get(i);**

**System.out.println(shape.toString());**

**}**

**}**

**});**

**// бутон за избор на цвят**

**Button chooseColor = new Button("Избор на цвят");**

**chooseColor.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**chosenColor = JColorChooser.showDialog(null, "Цветове", currentColor);**

**currentColor = chosenColor;**

**}**

**});**

**// добавяне на компонентите към панел, разположен в горната част на прозореца**

**Panel top = new Panel();**

**Label selectList = new Label("Избор на примитив:");**

**top.add(selectList);**

**top.add(shapeList);**

**top.add(displayShapes);**

**top.add(chooseColor);**

**add(top, BorderLayout.NORTH);**

**// слушател на мишката, необходим за изчертаване на примитивите**

**addMouseListener(new MouseAdapter() {**

**public void mousePressed(MouseEvent e) {**

**startX = e.getX();**

**startY = e.getY();**

**selectedShape = shapeList.getSelectedItem();**

**}**

**public void mouseReleased(MouseEvent e) {**

**endX = e.getX();**

**endY = e.getY();**

**if (selectedShape.equals(Line)) {**

**DrawableShape line = new Line(startX, startY, endX, endY, currentColor);**

**shapes.add(line);**

**} else if (selectedShape.equals(Rectangle)) {**

**x = FindMin(startX, endX);**

**y = FindMin(startY, endY);**

**width = FindMod(endX - startX);**

**height = FindMod(endY - startY);**

**DrawableShape rectangle = new Rectangle(x, y, width, height, currentColor);**

**shapes.add(rectangle);**

**} else if (selectedShape.equals(Oval)) {**

**x = FindMin(startX, endX);**

**y = FindMin(startY, endY);**

**width = FindMod(endX - startX);**

**height = FindMod(endY - startY);**

**DrawableShape oval = new Oval(x, y, width, height, currentColor);**

**shapes.add(oval);**

**}**

**repaint();**

**}**

**});**

**setVisible(true);**

**}**

**// обхождане на колекцията и извикване на метода drawShape в paint**

**public void paint(Graphics g) {**

**for (int i = 0; i < shapes.size(); i++) {**

**DrawableShape shape = shapes.get(i);**

**shape.drawShape(g);**

**}**

**}**

**// метод за намиране на минимална стойност**

**int FindMin(int a, int b) {**

**if (a < b) {**

**return a;**

**} else {**

**return b;**

**}**

**}**

**// метод за намиране на модул**

**int FindMod(int a) {**

**if (a < 0) {**

**return -a;**

**} else {**

**return a;**

**}**

**}**

**public static void main(String[] args) {**

**new GraphicsEditor();**

**}**

**}**